Guitar Delay Pedal Project

Background

- A delay is an audio effect which records an input signal to a storage medium, and plays it back after some period of time
- A guitar delay pedal is a device that delays the sound for some period of time before playing it back, according to the user specified length of time

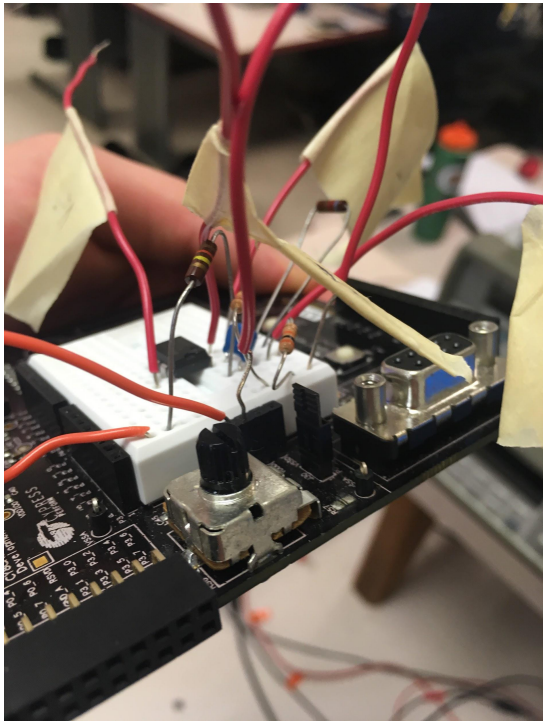
Challenge Elements

- Analog to Digital Converter (taking analog guitar input, converting to digital to read and utilize)
- Digital to Analog Converter (taking digital signal, converting to analog for output)
- Finite Impulse Response Digital Filter
- 2 Push Buttons

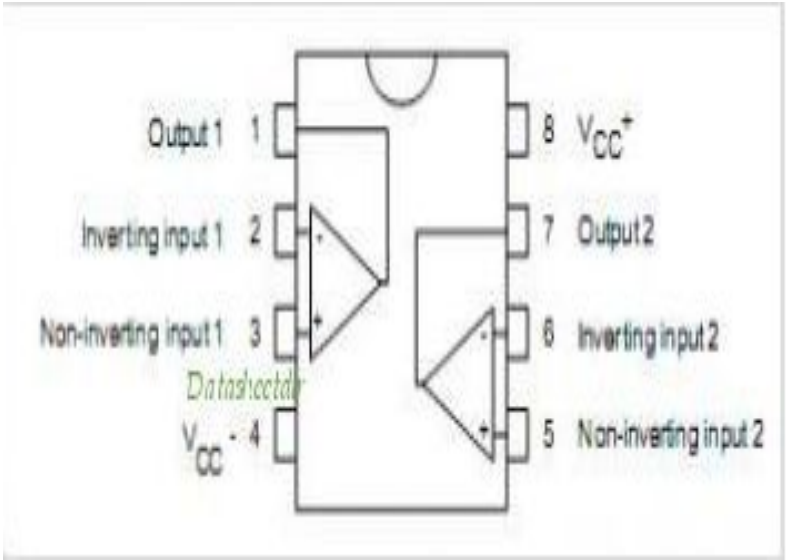
Objective

- Design a guitar pedal to take a guitar signal as an input, and output the delayed and echoed sound numerous times
- The pedal must implement a preamplifier in order to manage the negative voltage input
- The pedal must poll for updated input data in order to produce a continuous output
- The pedal must use an ADC in order to read and utilize the input

Current Model

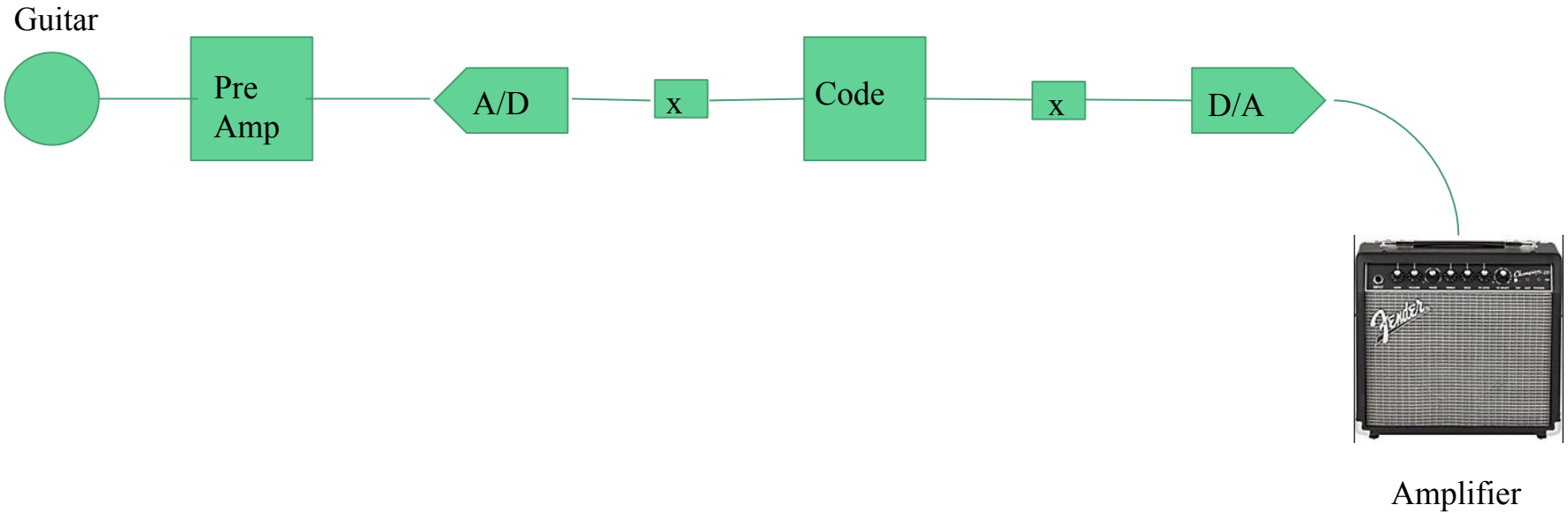


Physical Implementation of Circuit Design

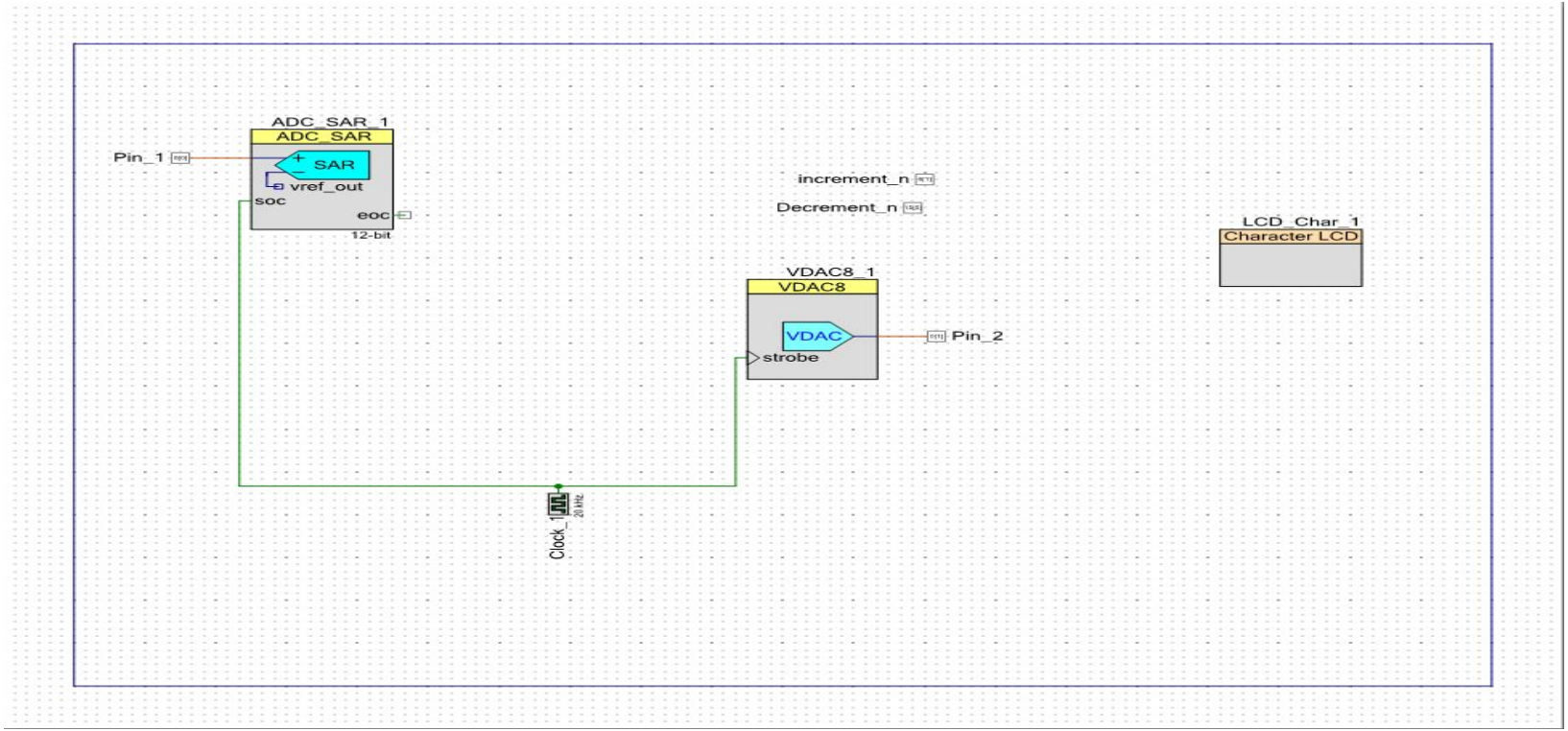


Dual Op-Amp (MC1458)

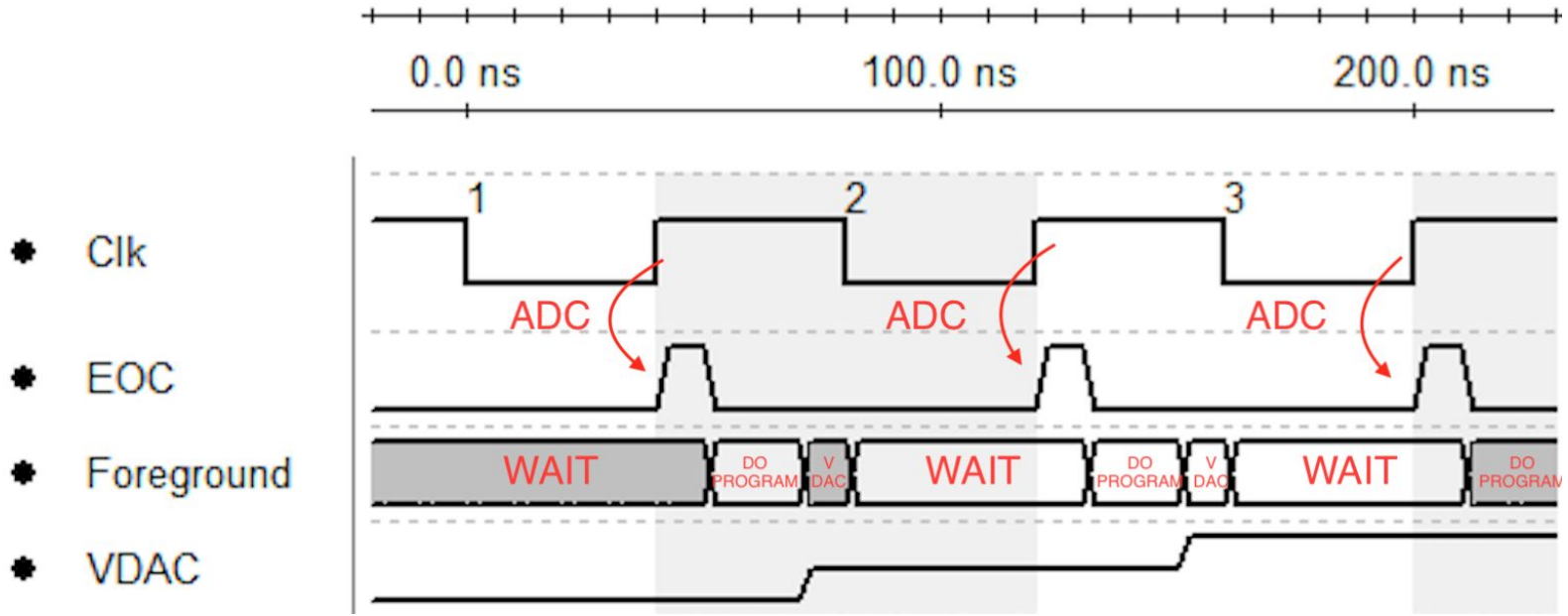
Block Diagram



PSoC Schematic



Timing Diagram



Current Model Code

```
int main()
{
    int maxs=20000;
    int sum=0;
    int windex=0;
    volatile uint16_t n=10;
    int delay=2000;
    int k;
    uint16_t x;
    uint16_t gsa [20000]={0};             //sample array initialized to 0


     CyGlobalIntEnable;              /* Enable global interrupts. */

     Clock_1_Start() ;
     ADC_SAR_1_Start();
     VDAC8_1_Start();
     LCD_Char_1_Start();
     LCD_Char_1_Position(0u, 0u);

     for(;;)
     {
     LCD_Char_1_ClearDisplay();
     LCD_Char_1_PrintNumber(n);                                 //outputs n to display
     CyDelay(500);

         if( Decrement_n_Read() == 0 )                          // if pressed
         {
             CyDelay(500);                                      //to make sure increment/decrement event registers only once per press
             n=n-1;                                             //decrement
         }
         else if( increment_n_Read() == 0)
         {
            CyDelay(500);
             n=n+1;                                             //Increment
         }


         ADC_SAR_1_IsEndConversion(ADC_SAR_1_WAIT_FOR_RESULT);//polling
         x=ADC_SAR_1_GetResult16();
         gsa[windex]=x;                                         //indexing sample array
         sum=0;
         for(k=0;k<n;++k)
         {
             sum+=gsa[(windex+maxs-k*delay)%maxs];              //circular buffer

         }
         sum=sum>>4;
         sum=sum/n;                                            //Prevents increasing amplitude with every echo
         VDAC8_1_SetValue(sum);
         windex=(windex+1)%(maxs);
         }
}
```

MATLAB Simulation

- MATLAB was used to debug and simulate the guitar delay pedal
- Utilized FIFO (first in first out), delay and echo

FIFO.m | Delay_Model.m | Echo.m

```
FIFO ('init');
fs = 20000;
f0=440;
 delay = 6000;
 n_echos = 6;
x=(1:10*fs);
x=exp(-1.*x./2000).*cos(2*pi()*fs/f0*(1+x/100000).*x);
plot(x(1:1000))
%x(1000:end) = 0;
%sound(x)

for n=1:20000 %assuming 10 seconds * 20,000
    y(n)=Echo(x(n),delay,n_echos);
end

    sound(y);
    plot(y);
```

FIFO.m | Delay_Model.m | Echo.m

```
function y = Echo(x, delay, n_echos )



   sum = 0;
  FIFO('write', x);
      for i = 1:n_echos
          sum = sum + FIFO('read', delay * i);
      end
   y = sum;
end
```

FIFO.m | Delay_Model.m | Echo.m

```
function result = FIFO (command, arg)
    if strcmp(command, 'read')
        result = read_fifo(arg);
    else
        if strcmp(command, 'write')
            write_fifo(arg);
        else
          if strcmp(command, 'init')
              init_fifo();
          end
        end
    end
end

function init_fifo()
global fifo_buf write_ptr MAX_DELAY; % globalizing variables
MAX_DELAY = 20000;%Max delay of 20kHz

fifo_buf = zeros(MAX_DELAY,1);%buffer = array of all zeros

write_ptr = 1;
end
function val = read_fifo(delay)% Val = fifo output
   global fifo_buf write_ptr MAX_DELAY;

    % Read delayed values
    % Delay read_ptr by an amount of samples, based on 'delay'
    read_ptr = mod((write_ptr - 2) - delay + MAX_DELAY, MAX_DELAY)+1 ;

    val = fifo_buf(read_ptr);

end

function write_fifo( val )%using output of read_fifo
global fifo_buf write_ptr MAX_DELAY;

    fifo_buf(write_ptr) = val;
    write_ptr = write_ptr + 1;%increment write_ptr by 1

    if (write_ptr > MAX_DELAY)%Once write_ptr is greater than max delay, restart
        write_ptr = 1;
    end
end
```

Results

- The results of our guitar delay pedal will be demonstrated in the video below: